AMENDMENTS TO THE CLAIMS

Please amend the present application as follows:

Claims

1.   (Original) A method of providing recovery from an error condition in a computer program, comprising the steps of:

parsing a source program for an error condition test;

detecting if an error condition test exists in said source program;

determining if error recovery is enabled when said error condition test is detected;

creating an error recovery flag code when said error condition test exists and said error recovery is enabled; and

inserting error recovery code in the computer program if error recovery is enabled.

2.   (Original) The method of claim 1, further comprising the steps of:

creating program abort code if said error condition test exists and error recovery is not enabled; and

generating code to perform said error condition test if said error condition test exists.

3.   (Currently amended) The method of claim [[3]] 2, further comprising the step of:

generating code to conditionally skip said program abort code and said error recovery flag code when said error condition test exists and said error recovery is not enabled.

4.   (Original) The method of claim 1, further comprising the steps of:

detecting if a call to a subroutine exists in said source program; and

creating an error recovery flag test code to test if said error recovery is enabled and said subroutine exists.

5.   (Original) The method of claim 4, further comprising the steps of:

generating code to conditionally skip said program abort code and said error recovery flag test code when said error condition test exists and said error recovery is not enabled.

2

6.   (Original) A system for providing recovery from an error condition in a computer program, said error recovery system comprising:

means for parsing a source program for an error condition test;

means for detecting if an error condition test exists in said source program;

means for determining if error recovery is enabled when said error condition test is detected;

means for creating an error recovery flag code when said error condition test exists and said error recovery is enabled; and

means for inserting error recovery code in the computer program if error recovery is enabled.

7.   (Original) The system of claim 6, further comprising:

means for creating program abort code if said error condition test exists and error recovery is not enabled; and

means for generating code to perform said error condition test if said error condition test exists.

8.   (Original) The system of claim 7, further comprising:

means for generating code to conditionally skip said program abort code and said error recovery flag code when said error condition test exists and said error recovery is not enabled.

9.   (Original) The system of claim 6, further comprising:

means for detecting if a call to a subroutine exists in said source program; and

means for creating an error recovery flag test code to test if said error recovery is enabled and said subroutine exists.

10.   (Original) The system of claim 9, further comprising:

means for generating code to conditionally skip said program abort code and said error recovery flag test code when said error condition test exists and said error recovery is not enabled.

3

11.   (Currently amended) An error recovery system providing recovery from an error condition in a computer program, comprising

a compiler configured to parse a source program;

error condition test logic configured to detect whether an error condition test exists in said source program;

determination logic configured to determine whether error recovery is enabled when said error condition test is detected;

error recovery flag generation logic configured to generate an error recovery flag code when said error condition test is detected and said error recovery is enabled; and

error recovery code logic configured to provide error ~~recover~~ recovery code that is inserted in said computer program if error recovery is enabled.

12.   (Original) The system of claim 11, further comprising:

abort code generator that generates abort code if said error condition test is detected and error recovery is not enabled; and

error condition test code generator that generates code to perform said error condition test if said error condition test is detected.

13.   (Original) The system of claim 12, further comprising:

conditional test code generator for generating code to conditionally skip said program abort code and said error recovery flag code when said error condition test is detected and said error recovery is not enabled.

14.   (Original) The system of claim 11, further comprising:

subroutine detection logic that detects if a call to a subroutine exists in said source program; and

error recovery flag test code generator that generates code to test if said error recovery is enabled and said subroutine exists.

4

15.   (Original) The system of claim 14, further comprising:

conditional subroutine test code generator for generating code to conditionally skip said program abort code and said error recovery flag code, when said error condition test is detected and said error recovery is not enabled.

16.   (Currently amended) A computer readable medium for providing recovery from an error condition in a computer program, comprising:

logic for parsing a source program for [[a]] an error condition test;

logic for detecting if an error condition test exists in said source program;

logic for determining if error recovery is enabled when said error condition test is detected;

logic for creating an error recovery flag code when said error condition test exists and said error recovery is enabled; and

logic for inserting error recovery code in the computer program if error recovery is enabled.

17.   (Original) The computer readable medium of claim 16, further comprising:

logic for creating program abort code if said error condition test exists and error recovery is not enabled; and

logic for generating code to perform said error condition test if said error condition test exists.

18.   (Original) The computer readable medium of claim 17, further comprising:

logic for generating code to conditionally skip said program abort code and said error recovery flag code when said error condition test exists and said error recovery is not enabled.

19.   (Original) The computer readable medium of claim 16, further comprising:

logic for detecting if a call to a subroutine exists in said source program; and

logic for creating an error recovery flag test code to test if said error recovery is enabled and said subroutine exists.

20.   (Original) The computer readable medium of claim 19, further comprising:

logic for generating code to conditionally skip said program abort code and said error

recovery flag test code when said error condition test exists and said error recovery is not

enabled.